## REMARKS

This Amendment is filed in response Office Action dated Aug. 4, 2009. As discussed below, the Applicant respectfully requests reconsideration. All objections and rejections are respectfully traversed.

Claims 1-10 and 12-34 are now pending in the application.

Claims 1-3, 5-6, 13,17, 20-24, 27-30 and 32-34 have been amended.

No new claims have been added.

### *Request for Interview*

The Applicant respectfully requests a telephonic interview to advance the prosecution of this case. The Applicant believes an interview will be most productive after the Examiner has had an opportunity to review this Amendment, but prior to the issue of the next Office Action. As the Applicant can not determine when the Examiner will have time to consider this Amendment, given PTO workload, the Applicant respectfully requests the Examiner contact the Applicant at 617-951-2500 when he reviews this Amendment so that a time convenient to the Examiner may be arranged for a telephonic interview.

### *Claim Rejections – 35 U.S.C. § 102*

At paragraphs 10-13 of the Office Action, claims 1, 13, 22 and 28 were rejected under 35 U.S.C. § 102(e) as over Williams, U.S. Patent No. 7,031,325 (hereinafter "Williams").

Applicant's claim 1, representative in part of claims 1, 13, 22 and 28, sets forth (emphasis added):

> 1. A method for modifying packet header data transferred from a context memory internal to a forwarding engine to an output buffer of the forwarding engine, the method comprising the steps of:

reading one or more instructions, by a processor of the forwarding engine, each instruction indicating an operation to modify the packet header data;

generating, in response to the one or more instructions, one or more commands wherein each command is associated with the operation to modify the packet header data;

placing the one or more commands in a data structure;

***initiating a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***; and

performing, by a device in the forwarding engine operating independently from the processor, the operations associated with the one or more commands contained in the data structure, to ***modify the packet header data*** as directed by the one or more commands ***while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine.***

Williams discusses a multiport switch that may operate "in accordance with both IEEE 802.1D and IEEE 802.1Q protocols." *See* col. 1, lines 64-66. IEEE 802.1Q protocol may require "received frames being modified before forwarding...." *See* col. 2, lines 21-26. First, "[t]he multiport switch stores the receive data frame in a receive FIFO buffer in receiver 205." *See* col. 10, lines 10-12 and Fig. 2, 205. Then "logic may transfer the data frame to external memory 170, via external memory interface 265." *See* col. 10, lines 14-17 and Fig. 1, 170 and Fig. 2, 265. A "transmit module may include dequeuing logic that obtains packets from the external memory 170 and stores the packets in the corresponding transmit FIFOs" *See* col. 5, lines 11-15 and Fig. 1, 170 and Fig. 3, 350A and 350B. "The dequeuing logic 350A may ... perform any necessary modification to the data frame while it transfers the data frame [from the external memory] into the transmit FIFO buffer, such as transmit FIFO buffer 350B...." *See* col. 12, lines 8-12 and Fig. 1, 170 and Fig. 3, 350A and 350B.

The Applicant respectfully urges that Williams does not teach or suggest the Applicant's claimed "***initiating a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***" and "***a device in the forwarding engine operating independently from the processor ...***

***modify the packet header data … while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***."

Previous forwarding engines have often employed a processor of the forwarding engine to modify portions of a packet header. However, employing the processor of the forwarding engine to modify portions of a packet header may not be an efficient use of the processor's resources. Depending on the amount of traffic being processed, the resources needed to perform the modifying may impact the processor's ability to perform other tasks, e.g., manage routing tables, respond to other devices in the network, etc. The Applicant has developed a novel technique that may address this shortcoming by offloading many packet header modification tasks generally performed by the processor of to a device in the forwarding engine that is operating independently from the processor. Specifically, the Applicant "***initiat[es] a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***" and then has "***a device in the forwarding engine operating independently from the processor … modify the packet header data … while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***."

Williams does not suggest what the Applicant claims. Rather than initiate, and then modify, packet header data while the packet header data is being transferred **from the context memory internal to the forwarding engine** to the output buffer of the forwarding engine, Williams stores packet data **in an external memory**, and transfers data there from. Thus Williams operates on a different transfer of data than what is claimed.

Williams makes clear that his data packet *are* moved to the external memory (Fig. 1, 170) upon receipt at the switch, and then is dequeued from the external memory 170 to his transmit FIFO buffer (Fig. 3, 350B) of the switch. *See* Williams col. 10, lines 14-17, col. 5, lines 11-15, col. 12, lines 8-12, and Fig. 1, 170 and Fig. 3, 350B. As shown in Williams Fig. 1, the external memory 170 is located outside of the multiport switch 180.

Such an external memory may not fairly be likened to a "***context memory internal to the forwarding engine.***" Thus, Williams may not fairly be interpreted as teaching a device in a forwarding engine operating independently from the processor "***modify the packet header data … while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine.***"

Accordingly, for the above reasons, the Applicant respectfully requests reconsideration and urges that Williams is legally insufficient to anticipate the present claims under 35 U.S.C. §102(e).

At paragraphs 14-31 of the Office Action, claims 1 – 4, 7, 9 – 15, 18, 20 – 25, 28 – 31 and 34 were rejected under 35 U.S.C. § 102(e) over Henderson et al., U.S. Publication No. 2004/004290 (hereinafter "Henderson").

Applicant's claim 1, representative in part of claims 1 – 4, 7, 9 – 15, 18, 20 – 25, 28 – 31 and 34, sets forth (emphasis added):

> 1. A method for modifying packet header data transferred from a context memory internal to a forwarding engine to an output buffer of the forwarding engine, the method comprising the steps of:
>
> reading one or more instructions, by a processor of the forwarding engine, each instruction indicating an operation to modify the packet header data;
>
> generating, in response to the one or more instructions, one or more commands wherein each command is associated with the operation to modify the packet header data;
>
> placing the one or more commands in a data structure;
>
> ***initiating a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***; and
>
> performing, by a device in the forwarding engine operating independently from the processor, the operations associated with the one or more commands contained in the data structure, to ***modify the packet header data*** as directed by the one or more commands ***while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***.

Henderson discloses a system and method for modifying received packets. A packet is received at a packet input unit 210. *See* paragraph 0030 and Fig. 2. "The packet input unit 210 stores the packet data into one or more free blocks of cache 230." *See* paragraph 0030 and Fig. 2. Under the direction of the packet processing controller (200), an editing unit (216) modifies the packet. *See* paragraph 0034. "Once all packet editing has been performed on a particular packet, a queue instruction [is] output by the editing unit 218" (emphasis added). *See* paragraph 0034, lines 40-43. Then, the service processor 210 groups the packets into queues stored in the cache 230. *See* paragraph 0035, lines 2-5. "Sometime after the packets have been added to a queue… the output scheduler 220 removes them [from cache 230] and sends them to the packet output unit 222." *See* paragraph 0036, lines 1-3.

The Applicant respectfully urges that Henderson does not teach or suggest the Applicant's claimed "***initiating a transfer of the packet header data from the context memory internal to the forwarding engine to the output buffer of the forwarding engine***" and "***modify the packet header data ... while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine.***"

While the Applicant **initiates a transfer** of packet header data from a context memory internal to a forwarding engine to an output buffer of the forwarding engine, and modifies the packet header data **while** the packet header data **is being transferred** between these two units, Henderson makes clear his packets are modified **before** any transfer of the packets is initiated from his cache 230 to his packet output unit 222.

Specifically, Henderson's editing unit 216 reads packets from Henderson's cache 230, modifies the packets, and returns the packets back to the same cache 230, organizing them into queues there. Then **"[s]ometime after the packets have been added to a queue**… the output scheduler 220 removes them [from cache 230] and sends them to the packet output unit 222." *See* Henderson paragraph 0036, lines 1-3 (emphasis added).

Henderson does not suggest performing any modification to a packet while the packet is in the process of being sent/transferred from his cache 230 to the output unit 222. As such, Henderson cannot fairly be interpreted as teaching modifying packet header data **while** the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine. If anything, Henderson teaches away from this type of operation, by stating packets are sent/transferred "**sometime after**" any modifications are complete.

At paragraph 7 of the Office Action, the Examiner suggests that "Henderson discloses each packet that arrives at a services processor is assigned a packet context that contains information about the packet, state information related to the packet, and actions to be done to the packet ... Information within the state tables is often modified in real time as the packets are processed", apparently quoting from paragraphs 0007 and 0008 of Henderson. Yet updating "**state tables**" in real time as packets are processed is quite different from modifying **packet header data while the packet header data is being transferred from the context memory internal to the forwarding engine to the output buffer of the forwarding engine.** Henderson's "state tables" do not store packet header data. Rather, Henderson makes clear that his "state tables" store state information stores other types of information, such as information descriptive of cumulative packet flows. For example, Henderson explains at paragraph 0068 that "the state table 720 stores information about the ports of the switch 106. One or more of the information fields in the state table 720 stores the number of packets that have been sent from each output port. When a packet 301 is to be sent out through particular output port, the state table 720 is updated to reflect that another packet has been sent from that port." Accordingly, Henderson's updating of state information in his state tables, bears little relation to how packet header data is modified.

Therefore, for the above reasons, the Applicant respectfully requests reconsideration and urges that Henderson is legally insufficient to anticipate the present claims under 35 U.S.C. §102(e).

***Claim Rejections – 35 U.S.C. § 103***

At paragraphs 32-35 of the Office Action, claims 5, 16, 26, and 32 were rejected under 35 U.S.C. § 103(a) over Henderson in view of Ueno, U.S. Publication No. 2002/0009050 (hereinafter "Ueno").

The Applicant respectfully notes that claims 5, 16, 26, and 32 are dependent claims that depend from independent claims that are believed to be in condition for allowance. Accordingly, claims 5, 16, 26, and 32 are believed to be in condition for allowance.

At paragraphs 36-38 of the Office Action, claims 8 and 19 were rejected under 35 U.S.C. § 103(a) over Henderson in view of Deforche et al., U.S. Publication No. 2005/0232303 (hereinafter "Deforche").

Applicant respectfully notes that claims 8 and 19 are dependent claims that depend from independent claims that are believed to be in condition for allowance. Accordingly, claims 8 and 19 are believed to be in condition for allowance.

In the event that the Examiner deems personal contact desirable in disposition of this case, the Examiner is encouraged to call the undersigned attorney at (617) 951-2500.

In summary, all the independent claims are believed to be in condition for allowance and therefore all dependent claims that depend there from are believed to be in condition for allowance. The Applicant respectfully solicits favorable action.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

__/James A. Blanchette/_____
James A. Blanchette
Reg. No. 51,477
CESARI AND MCKENNA, LLP
88 Black Falcon Avenue
Boston, MA 02210-2414
(617) 951-2500